

Credit Name: CSE 3120 Object-Oriented Programming 1

Assignment Name: Lunch Order Mastery

How has your program changed from planning to coding to now? Please explain?

PLANNING:

I plan to create two classes, one for the main Lunch Order program and another to calculate the food facts (carbs, fat, fibre) and total price.

Lunch Order: Declare variables/scanner, create objects linking the FoodInfo Class, prompt the user for the amount of each food item and store the number, then print food facts. The stored number will be used to calculate and print the order total.

FoodInfo: This will hold the methods that return carbs, fat, fiber, and the item price.

CODING:

1. Lunch Order class

```
public static void main(String[] args) {  
  
    //Initialize variables used for calculating ordertotal  
    //variables store amount of each item ordered  
    int numham;  
    int numsalad;  
    int numff;  
    int numsoda;  
  
    //Prepare scanner for user input  
    Scanner input = new Scanner(System.in);  
  
    //Decimal format used when printing order total later  
    DecimalFormat formatting = new DecimalFormat("#.##");  
}
```

Variables that will store the amount of each food item ordered by the user are declared.

Scanner is prepared for user input.

The decimal format is set to two decimal places.

(It will be used when calculating the order total later on in the program).

2. FoodInfo Class

```
// declaring variables to store the price, carbs, fat, and fiber  
private double itemprice;  
private int itemcarb;  
private int itemfat;  
private int itemfiber;  
  
//constructor method with parameters price, carbs, fat, and fiber  
//initialize the variables above  
public FoodInfo(double price, int carbs, int fat, int fiber) {  
    itemprice = price;  
    itemcarb = carbs;  
    itemfat = fat;  
    itemfiber = fiber;  
}
```

Variables are declared in this class that will store the price, carbs, fat, and fiber amounts for different food items.

A constructor method with the parameters price, carbs, fat, and fiber will be used when creating objects in the main Lunch Order class.

Inside the constructor method, the previously declared variables are initialized to equal the parameter variables.

3. FoodInfo Class

```
// method gives the price of item
public double TPrice() {
    return itemprice;
}

//method gives amount of carbs for the item
public int GCarb() {
    return itemcarb;
}

//method gives amount of fat for the item
public int GFat() {
    return itemfat;
}

//method gives amount of fiber for the item
public int GFiber() {
    return itemfiber;
}
```

TPrice method returns the price for an item

GCarb method returns carb value of an item

GFat method returns fat value of an item

GFiber method returns fiber value of an item

4. Lunch Order Class

```
//linking the food item variables from the Lunch Order class to the price, carbs, fat, fiber;
// Ex. FoodInfo soda = new FoodInfo(price, carbs, fat, fiber);
FoodInfo hamburger = new FoodInfo(1.85, 33, 9, 1);
FoodInfo salad = new FoodInfo(2.00, 11, 1, 5);
FoodInfo ff = new FoodInfo(1.30, 36, 11, 4);
FoodInfo soda = new FoodInfo(0.95, 38, 0, 0);

//Intro message welcoming users
System.out.println("Welcome! This program helps you make a lunch order.");
```

Hamburger, salad, ff, and soda objects are created, with parameters in the following order: (price, carbs, fat, fiber). This connects back to step two, where I created a constructor method in the FoodInfo class with the same parameters.

The introduction message is printed.

5. Lunch Order Class

```
//prompt user for number of hamburgers
System.out.println("Enter number of hamburgers: ");
//store user value in the numham variable
numham = input.nextInt();
//print hamburger facts
System.out.println("Each hamburger has " + hamburger.GCarb() +
    "g of carbs, " + hamburger.GFat() + " g of fat, and " + hamburger.GFiber() + "g of fiber");

//prompt user for number of salads
System.out.println("Enter number of Salads: ");
//store user value in the numsalad variable
numsalad = input.nextInt();
//print salad facts
System.out.println("Each salad has " + salad.GCarb() +
    "g of carbs, " + salad.GFat() + " g of fat, and " + salad.GFiber() + "g of fiber");

//prompt user for the number of French Fries
System.out.println("Enter number of French Fries: ");
//store user value in the numff variable
numff= input.nextInt();
//print French Fries facts
System.out.println("Each order of French Fries has " + ff.GCarb() +
    "g of carbs, " + ff.GFat() + " g of fat, and " + ff.GFiber() + "g of fiber");

//prompt user for the number of sodas
System.out.println("Enter number of Sodas: ");
//store user value in the numsoda variable
numsoda = input.nextInt();
//print soda facts
System.out.println("Each soda has " + soda.GCarb() +
    "g of carbs, " + soda.GFat() + " g of fat, and " + soda.GFiber() + "g of fiber");
```

User is prompted for number of hamburgers, salads, french fries, and sodas they would like to purchase.

These numbers are then stored in the appropriate variables.

The food facts are then printed for each food item. To print the food facts such as carbs, fat, and fiber, the appropriate objects are linked to the appropriate methods in the FoodInfo class.

For example, in hamburger.Gcarb() the object is hamburger, and .GCarb() is the method that returns the carb amount for the hamburger (found in FoodInfo Class.)

6. Lunch Order Class

```
//print the order total
//use the decimal formatting declared at the start of the program
// multiplying number of item times the item price, then adding all of the item prices tog
System.out.println("Your order total is: $" + formating.format(numham*hamburger.TPrice() +
    numsalad*salad.TPrice() + numff*ff.TPrice() + numsoda*soda.TPrice()));
```

Printing the total \$ amount using the formatting. I calculated the total by multiplying number of item * the price of the item for all items.

End Of Program!